Filter
< Architecture Center home
What's new
Fundamentals
Content overview
Well-Architected Framework
Overview
What's new
Operational excellence
Security, privacy, and compliance
Reliability
Cost optimization
Performance optimization
Sustainability
AI and ML perspective
View on one page
Deployment archetypes
Landing zone design
Enterprise foundations blueprint
AI and machine learning
Application development
Big data and analytics
Databases
Hybrid and multicloud

Google Cloud Well-Architected Framework

Last reviewed 2024-10-11 UTC

To view all of the content in the Well-Architected Framework on a single page or to to get a PDF output of the content, see View on one page.

The Well-Architected Framework provides recommendations to help architects, developers, administrators, and other cloud practitioners design and operate a cloud topology that's secure, efficient, resilient, high-performing, and cost-effective.

A cross-functional team of experts at Google validates the recommendations in the Well-Architected Framework. The team curates the Well-Architected Framework to reflect the expanding capabilities of Google Cloud, industry best practices, community knowledge, and feedback from you. For a summary of the significant changes to the Well-Architected Framework, see What's new.

The Well-Architected Framework is relevant to applications built for the cloud and for workloads migrated from on-premises to Google Cloud, hybrid cloud deployments, and multi-cloud environments.

Well-Architected Framework pillars and perspectives

The Well-Architected Framework is organized into five pillars, as shown in the following diagram. We also provide cross-pillar perspectives that focus on recommendations for selected domains, industries, and technologies like AI and machine learning (ML).

Pillars

Operational excellence

Efficiently deploy, operate, monitor, and manage your cloud workloads.

Security, privacy, and compliance

Maximize the security of your data and workloads in the cloud, design for privacy, and align with regulatory requirements and standards.

Reliability

Design and operate resilient and highly available workloads in the cloud.

Cost optimization

Maximize the business value of your investment in Google Cloud.

Performance optimization

Design and tune your cloud resources for optimal performance.

Perspectives

AI and ML

A cross-pillar view of recommendations that are specific to AI and ML workloads.

Core principles

Before you explore the recommendations in each pillar of the Well-Architected Framework, review the following core principles:

Design for change

No system is static. The needs of its users, the goals of the team that builds the system, and the system itself are constantly changing. With the need for change in mind, build a development and production process that enables teams to regularly deliver small changes and get fast feedback on those changes. Consistently demonstrating the ability to deploy changes helps to build trust with stakeholders, including the teams responsible for the system, and the users of the system. Using DORA's software delivery metrics can help your team monitor the speed, ease, and safety of making changes to the system.

Document your architecture

When you start to move your workloads to the cloud or build your applications, lack of documentation about the system can be a major obstacle. Documentation is especially important for correctly visualizing the architecture of your current deployments.

Quality documentation isn't achieved by producing a specific amount of documentation, but by how clear content is, how useful it is, and how it's maintained as the system changes.

A properly documented cloud architecture establishes a common language and standards, which enable cross-functional teams to communicate and collaborate effectively. The documentation also provides the information that's necessary to identify and guide future design decisions. Documentation should be written with your use cases in mind, to provide context for the design decisions.

Over time, your design decisions will evolve and change. The change history provides the context that your teams require to align initiatives, avoid duplication, and measure performance changes effectively over time. Change logs are particularly valuable when you onboard a new cloud architect who is not yet familiar with your current design, strategy, or history.

Analysis by DORA has found a clear link between documentation quality and organizational performance — the organization's ability to meet their performance and profitability goals.

Simplify your design and use fully managed services

Simplicity is crucial for design. If your architecture is too complex to understand, it will be difficult to implement the design and manage it over time. Where feasible, use fully managed services to minimize the risks, time, and effort associated with managing and maintaining baseline systems.

If you're already running your workloads in production, test with managed services to see how they might help to reduce operational complexities. If you're developing new workloads, then start simple, establish a minimal viable product (MVP), and resist the urge to over-engineer. You can identify exceptional use cases, iterate, and improve your systems incrementally over time.

Decouple your architecture

Research from DORA shows that architecture is an important predictor for achieving continuous delivery. Decoupling is a technique that's used to separate your applications and service components into smaller components that can operate independently. For example, you might separate a monolithic application stack into individual service components. In a loosely coupled architecture, an application can run its functions independently, regardless of the various dependencies.

A decoupled architecture gives you increased flexibility to do the following:

- Apply independent upgrades.
- Enforce specific security controls.
- Establish reliability goals for each subsystem.
- Monitor health.
- Granularly control performance and cost parameters.

You can start the decoupling process early in your design phase or incorporate it as part of your system upgrades as you scale.

Use a stateless architecture

A stateless architecture can increase both the reliability and scalability of your applications.

Stateful applications rely on various dependencies to perform tasks, such as local caching of data. Stateful applications often require additional mechanisms to capture progress and restart gracefully. Stateless applications can perform tasks without significant local dependencies by using shared storage or cached services. A stateless architecture enables your applications to scale up quickly with minimum boot dependencies. The applications can withstand hard restarts, have lower downtime, and provide better performance for end users.

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License, and code samples are licensed under the Apache 2.0 License. For details, see the Google Developers Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2024-10-11 UTC.